IN THE CLAIMS:

1. (Currently amended) A method for monitoring performance of a program being executed using symmetric multiprocessing (SMP) functionality comprising:

executing a native code routine;

executing a first thread of the native code routine on a first symmetric multiprocessing (SMP) processor;

ascertaining first profile information, wherein the first profile information is processor profile information for the first SMP processor that identifies metrics associated with the execution of the first thread on the first SMP processor and is maintained by an operating system kernel;

updating first thread profile information, maintained by a profiler application, with the first profile information;

executing the first thread of the native code routine on a second SMP processor;

ascertaining second profile information, wherein the second profile information is processor profile information for the second SMP processor that identifies metrics associated with the execution of the first thread on the second SMP processor and is maintained by the operating system kernel; and

updating the first thread profile information, maintained by the profiler application, with the second profile information.

2. (Original) The method recited in claim 1 above, further comprises:

executing a second thread of the native code routine on the first SMP processor;

ascertaining third profile information; and

updating second thread profile information with the third profile information.

3. (Canceled)

4. (Original) The method recited in claim 1 above, wherein the first profile information relates to the execution of the first thread on a virtual machine.

5. (Original) The method recited in claim 1 above, wherein ascertaining first profile information further comprises:

retrieving processor accumulated profile information;

retrieving processor last profile information; and

calculating the first profile information by comparing the processor accumulated profile information and the processor last profile information.

6. (Currently amended) The method recited in claim 5 above further comprises:

resetting the processor last profile information by replacing the [[last]] processor last profile information with the processor accumulated profile information.

7. (Original) The method recited in claim 1 above, wherein ascertaining first profile information further comprises:

accessing a first processor data area containing first processor accumulated profile information; and

updating the first processor accumulated profile information with virtual machine profile information, wherein the first profile information is calculated from the first processor accumulated profile information.

8. (Original) The method recited in claim 7 above, wherein updating the first processor accumulated profile information with virtual machine profile information is performed by a virtual machine.

9. (Original) The method recited in claim 7 above, wherein updating the first processor accumulated profile information with virtual machine profile information is performed by a virtual machine further comprises:

receiving a request to update the first processor accumulated profile information with virtual machine profile information;

accessing a first processor data area containing first processor accumulated profile information; and updating the first processor accumulated profile information with virtual machine profile information, wherein updating the first processor accumulated profile information with virtual machine profilc information is performed by an operating system kernel.

10. (Original) The method recited in claim 1 above, wherein ascertaining first profile information further comprises:

receiving virtual machine profile information;

updating processor accumulated profile information with virtual machine profile information;

retrieving processor last profile information; and

calculating the first profile information by comparing the processor accumulated profile information and the processor last profile information.

11. (Original) The method recited in claim 7 above, wherein the first processor data area is a predefined data area.

12. (Original) The method recited in claim 7 above, wherein the first processor data area is allocated for first processor accumulated profile information at initialization.

13. (Original) The method recited in claim 1 above, wherein the first processor accumulated profile comprises one of allocation bytes, allocation objects, time, live object and live bytes.

14. (Currently amended) A method for monitoring performance of a program being executed by symmetric multiprocessing (SMP) functionality comprising:

launching a current thread of a native code routine on a first symmetric multiprocessing (SMP) processor;

stopping [[the]] a last thread on the first SMP processor;

accessing a first SMP processor data area;

retrieving first SMP processor accumulated profile information;

retrieving first SMP processor last accumulated profile information;

ascertaining first profile information from the first SMP processor accumulated profile information and the first SMP processor last accumulated profile information;

updating first thread profile information with the first profile information;

setting first SMP processor last accumulated profile information in the first SMP processor data area equal to the first SMP processor accumulated profile information; and

starting the first thread on the first SMP processor.

15. (Currently amended) A method for monitoring performance of a program being executed by symmetric multiprocessing (SMP) functionality comprising:

receiving a request for current thread profile information:

retrieving SMP processor accumulated profile information from each SMP processor's data area;

retrieving SMP processor last accumulated profile information from each processor's data area;

ascertaining profile information from the SMP processor accumulated profile information and the SMP processor last accumulated profile information for each thread running on a processor;

updating thread profile information with the profile information for each thread running on a processor;

setting each SMP processor last accumulated profile information in [[the]] a respective SMP processor's data area equal to the SMP processor accumulated profile information for that SMP processor; and

transferring the thread profile information to the requester.

16-20. (Canceled)

21. (Currently amended) A data processing system for monitoring performance of a program being executed by symmetric multiprocessing (SMP) functionality comprising:

executing means for executing a native code routine;

executing means for ~~executing means for~~ executing a first thread of the native code routine on a first symmetric multiprocessing (SMP) processor;

ascertaining means for ascertaining first profile information, wherein the first profile information is processor profile information for the first SMP processor that identifies metrics associated with the execution of the first thread on the first SMP processor and is maintained by an operating system kernel;

updating means for updating first thread profile information, maintained by a profiler application, with the first profile information;

executing means for executing the first thread of the native code routine on a second SMP processor;

~~a second thread of the native code routine; and~~

ascertaining means for ascertaining second profile information, wherein the second profile information is processor profile information for the second SMP processor that identifies metrics associated with the execution of the first thread on the second SMP processor and is maintained by the operating system kernel; and

updating means for updating first thread profile information, maintained by the profiler application, with the second profile information.

22. (Original) The system recited in claim 21 above, further comprises:

executing means for executing a second thread of the native code routine on the first SMP processor;

ascertaining means for ascertaining third profile information; and

updating means for updating second thread profile information with the third profile information.

23. (Canceled)

24. (Original) The system recited in claim 21 above, wherein the first profile information relates to the execution of the first thread on a virtual machine.

25. (Original) The system recited in claim 21 above, wherein the ascertaining means for ascertaining first profile information further comprises:

retrieving means for retrieving processor accumulated profile information;

retrieving means for retrieving processor last profile information; and

calculating means for calculating the first profile information by comparing the processor accumulated profile information and the processor last profile information.

26. (Original) The system recited in claim 25 above further comprises:

resetting means for resetting the processor last profile information by replacing the last processor profile information with the processor accumulated profile information.

27. (Original) The system recited in claim 21 above, wherein the ascertaining means for ascertaining first profile information further comprises:

accessing means for accessing a first processor data area containing first processor accumulated profile information; and

updating means for updating the first processor accumulated profile information with virtual machine profile information, wherein the first profile information is calculated from the first processor accumulated profile information.

28. (Original) The system recited in claim 27 above, wherein the updating means for updating the first processor accumulated profile information with virtual machine profile information is performed by a virtual machine.

29. (Original) The system recited in claim 27 above, wherein the updating means for updating the first processor accumulated profile information with virtual machine profile information is performed by a virtual machine further comprises:

receiving means for receiving a request to update the first processor accumulated profile information with virtual machine profile information;

accessing means for accessing a first processor data area containing first processor accumulated profile information; and

updating means for updating the first processor accumulated profile information with virtual machine profile information, wherein updating the first processor accumulated profile information with virtual machine profile information is performed by an operating system kernel.

30. (Currently amended) The system recited in claim 21 above, wherein the ascertaining means for ascertaining first profile information further comprises:

receiving means for receiving virtual machine profile information;

updating means for updating processor accumulated profile information with virtual machine profile information;

retrieving means for ~~retrieving means for~~ retrieving processor last profile information; and

calculating means for calculating the first profile information by comparing the processor accumulated profile information and the processor last profile information.

31. (Original) The system recited in claim 27 above, wherein the first processor data area is a predefined data area.

32. (Original) The system recited in claim 27 above, wherein the first processor data area is allocated for first processor accumulated profile information at initialization.

33. (Original) The system recited in claim 21 above, wherein the first processor accumulated profile comprises one of allocation bytes, allocation objects, time, live object and live bytes.

34. (Currently amended) A computer program product for monitoring performance of a program being executed by symmetric multiprocessing (SMP) functionality comprising:

executing instructions for executing a native code routine;

executing instructions for executing a first thread of the native code routine on a first symmetric multiprocessing (SMP) processor;

ascertaining instructions for ascertaining first profile information, wherein the first profile information is processor profile information for the first SMP processor that identifies metrics associated with the execution of the first thread on the first SMP processor and is maintained by an operating system kernel;

updating instructions for updating first thread profile information, maintained by a profiler application, with the first profile information;

executing instructions for executing the first thread of the native code routine on a second SMP processor;

~~a second thread of the native code routine; and~~

ascertaining instructions for ascertaining second profile information, wherein the second profile information is processor profile information for the second SMP processor that identifies metrics associated with the execution of the first thread on the second SMP processor and is maintained by the operating system kernel; and

updating instructions for updating first thread profile information, maintained by the profiler application, with the second profile information.

35. (Currently amended) The computer program product recited in claim 34 above, further comprises:

executing means for executing a second thread of the native code routine on the first SMP processor;

ascertaining means for ~~ascertaining means for~~ ascertaining third profile information; and

updating means for updating second thread profile information with the third profile information.

36. (Canceled)

37. (Original) The computer program product recited in claim 34 above, wherein the first profile information relates to the execution of the first thread on a virtual machine.

38. (Original) The computer program product recited in claim 34 above, wherein the ascertaining instructions for ascertaining first profile information further comprises:

retrieving instructions for retrieving processor accumulated profile information;

retrieving instructions for retrieving processor last profile information; and calculating instructions for calculating the first profile information by comparing the processor accumulated profile information and the processor last profile information.

39. (Original) The computer program product recited in claim 38 above further comprises:

resetting instructions for resetting the processor last profile information by replacing the last processor profile information with the processor accumulated profile information.

40. (Original) The computer program product recited in claim 34 above, wherein the ascertaining instructions for ascertaining first profile information further comprises:

accessing instructions for accessing a first processor data area containing first processor accumulated profile information; and

updating instructions for updating the first processor accumulated profile information with virtual machine profile information, wherein the first profile information is calculated from the first processor accumulated profile information.

41. (Original) The computer program product recited in claim 40 above, wherein the updating instructions for updating the first processor accumulated profile information with virtual machine profile information is performed by a virtual machine.

42. (Original) The computer program product recited in claim 40 above, wherein the updating instructions for updating the first processor accumulated profile information with virtual machine profile information is performed by a virtual machine further comprises:

receiving instructions for receiving a request to update the first processor accumulated profile information with virtual machine profile information;

accessing instructions for accessing a first processor data area containing first processor accumulated profile information; and

updating instructions for updating the first processor accumulated profile information with virtual machine profile information, wherein updating the first processor accumulated profile information with virtual machine profile information is performed by an operating system kernel.

43. (Original) The computer program product recited in claim 34 above, wherein the ascertaining instructions for ascertaining first profile information further comprises:

receiving instructions for receiving virtual machine profile information;

updating instructions for updating processor accumulated profile information with virtual machine profile information;

retrieving instructions for retrieving processor last profile information; and

calculating instructions for calculating the first profile information by comparing the processor accumulated profile information and the processor last profile information.

44. (Original) The computer program product recited in claim 40 above, wherein the first processor data area is a predefined data area.

45. (Original) The computer program product recited in claim 40 above, wherein the first processor data area is allocated for first processor accumulated profile information at initialization.

46. (Original) The computer program product recited in claim 34 above, wherein the first processor accumulated profile comprises one of allocation bytes, allocation objects, time, live object and live bytes.

47. (New) The method of claim 1, wherein the processor profile information for the first SMP processor and the processor profile information for the second SMP processor are stored in per-processor data areas for each metric being tracked for each processor, and wherein the per-processor data areas are allocated by the operating system kernel at initialization of monitoring the performance of the program being executed.

48. (New) The method of claim 5, wherein the processor accumulated profile information is a total accumulated amount of a particular metric for a processor since

initialization of monitoring the performance of the program, and wherein the processor last accumulated profile information is an accumulated amount of a particular metric for a processor at a last time that the first thread profile information was updated.

49. (New) The method of claim 14, wherein the first SMP processor accumulated profile information is a total accumulated amount of a particular metric for the first SMP processor since initialization of monitoring the performance of the program, and wherein the first SMP processor last accumulated profile information is an accumulated amount of a particular metric for the first SMP processor at a last time that the first thread profile information was updated.

50. (New) The method of claim 15, wherein the SMP processor accumulated profile information is a total accumulated amount of a particular metric for a corresponding SMP processor since initialization of monitoring the performance of the program, and wherein the SMP processor last accumulated profile information is an accumulated amount of a particular metric for a corresponding SMP processor at a last time that the thread profile information was updated.

51. (New) A method of monitoring performance of a program being executed by a symmetric multiprocessing (SMP) system, comprising:

using an operating system kernel to maintain a first set of metrics associated with execution of a thread on a plurality of processors in the SMP system, wherein per-processor data areas are allocated by the operating system kernel, and associated with corresponding processors in the plurality of processors, to store metrics for the corresponding processors, and wherein the first set of metrics are at a processor-level;

using a profiler application to maintain a second set of metrics associated with threads of execution of the program, wherein the second set of metrics are at a thread level;

updating the second set of metrics based on the first set of metrics in response to a request from a requestor to update the second set of metrics to thereby generate an update second set of metrics; and providing the updated second sct of metrics to the requestor.